Mathematical Methods in Operations Research 2005
Special issue in honor of *Arie Hordijk*

Dynamic Programming in Real Life: A Two-Person Dice Game

Henk Tijms[1], **Jan van der Wal**[2]

[1] Department of Econometrics, Vrije Universiteit, Amsterdam, The Netherlands. E-mail: tijms@feweb.vu.nl

[2] Department of Quantitative Economics, Faculty of Economics and Econometrics, University of Amsterdam and Department of Mathematics and Computing Science, Eindhoven University of Technology, Eindhoven, The Netherlands. E-mail: jan.v.d.wal@tue.nl

Received: January 2005 / Revised version: April 2005

Abstract Dynamic programming can solve a broad range of optimization problems. In the seventies and eighties of the last century the fundamentals of dynamic programming were developed. In this paper we present a real-world application of dynamic programming and stochastic game theory. This problem offers challenging questions of a general nature.

1 Introduction

Arie Hordijk has worked in many fields, among them dynamic programming. Dynamic programming is a branch of applied mathematics, which greatly developed in the period 1970-1985. Arie Hordijk made path-breaking contributions to the field in that period (and also afterwards 1985), first as member of the Amsterdam group and later as chairholder in Leiden. His thesis "Dynamic Programming and Markov Potential Theory" published in 1974 is a milestone in the field, see [2]. The Amsterdam group, the Eindhoven group as well as groups in Germany were active in the field of dynamic programming and stochastic games in the seventies and the eighties of the last century. These groups had joint conferences, amongst others in a castle in Rheda, and those meetings were quite stimulating for the many contributions made to the field. The authors of this paper belonged in those days to the Amsterdam group and the Eindhoven group, respectively. It is therefore a pleasure to contribute to this special issue, and, particularly, to an interesting problem in dynamic programming and stochastic games. This problem is a real-world problem, which might seem of recreational nature at first sight, but offers many challenging questions of a general nature. Questions we can only partially answer in this paper.

The problem deals with a real-world situation arising in the final of an American TV show. At the end of the show the two remaining contestants have to play a two-person dice game. The contestants each sit behind a panel with a battery of buttons numbered as $1, 2, \ldots, 10$. In each stage of the game, the contestants must simultaneously press one of the buttons, where the contestants cannot observe each other's decision. The number on the button pressed by the contestant is the number of dice that are thrown for the contestant. For each contestant the score of the throw for that contestant is added to his/her total, provided that none of the dice in that throw showed the outcome 1; otherwise no points are added to the current total of the candidate. The candidate who first reaches a total of G points is the winner. In case both candidates reach the goal of G points in the same move, the winner is the candidate who has the largest total. In case these totals are equal, the game is called a tie. At each stage of the game both candidates have full information about his/her own current total and the current total of the opponent. The formulation of the game will be such that it is zero-sum and stochastic. What is the optimal strategy looking like? Do random actions appear or not? And if so, when?

2 Some preliminaries

Let us first look at the distribution of the number of points earned in a single throw with d dice. Define the random variable Y_d as

$Y_d :=$ the number of points added to a contestant's total, throwing d dice.

Letting the random variable X_i denote the number of pips shown by the i-th dice, Y_d equals $X_1 + \cdots + X_d$ if none of $X_1, \ldots, X_d$ equals 1 and Y_d is 0 otherwise. The random variables $X_1, \ldots, X_d$ are independent and identically distributed. Moreover, given that X_i is not 1, the conditional distribution of X_i is the uniform distribution on the integers $2, 3, \ldots, 6$. This conditional distribution has expected value 4. The probability of getting not a single 1 in a throw of d dice is $(\frac{5}{6})^d$. Elementary calculations next show that

$$E(Y_d) = \left(\frac{5}{6}\right)^d 4d \quad \text{and} \quad \operatorname{var}(Y_d) = \left(\frac{5}{6}\right)^d (16d^2 + 2d) - \left(\left(\frac{5}{6}\right)^d 4d\right)^2 .$$

The maximum of $E(Y_d)$ is easily found by looking at the difference between $E(Y_{d+1})$ and $E(Y_d)$:

$$E(Y_{d+1}) - E(Y_d) = \left(\frac{5}{6}\right)^d 4 \left(\frac{5}{6}(d+1) - d\right) .$$

The difference is positive for $d < 5$, is zero for $d = 5$, and is negative for $d > 5$. Hence $E(Y_d)$ is maximized by taking d equal to 5 or 6.

Remark. A more intuitive reasoning is the following. Given that you already have put d dice in your hand, should you pick up another one? If one of the previous dice will give a 1 it is irrelevant what you do. So assume none of the other dice will give a 1. Then on the average, every one of them will contribute 4 points. So, in this situation with probability 1/6 you loose 4d points and with probability 5/6 you win another 4 points. This is essentially the same comparison we made before.

The following table gives the probability $q_0^{(d)}$ together with the mean μ_d and standard deviation σ_d of the random variable Y_d for various values of d.

d	μ_d	σ_d	$q_0^{(d)}$	d	μ_d	σ_d	$q_0^{(d)}$
1	3.3333	1.9720	0.8333	7	7.8143	12.7139	0.2791
2	5.5556	4.0445	0.6944	8	7.4422	13.6559	0.2326
3	6.9444	6.2113	0.5787	9	6.9770	14.3521	0.1938
4	7.7160	8.2327	0.4823	10	6.4602	14.8292	0.1615
5	8.0376	10.0084	0.4019	20	2.0867	12.7917	0.0261
6	8.0376	11.5029	0.3349	30	0.5055	7.7885	0.0042

Table 1 Mean, standard deviation and $q_0^{(d)}$ for one throw.

As we see, if we only look at the mean, the optimal number of dice is 5 or 6 for the situation of a single move. But as the standard deviation shows, throwing with 5 or 6 dice is not the same. With 6 dice the throw will be more 'risky'. If you quickly need a lot of points, then you have to take a risk and throws with 7 or more dice come in the picture.
Next, we discuss how to compute the probability distribution of Y_d. For any $d \geq 1$, let

$$q_i^{(d)} = P(Y_d = i) \quad \text{and} \quad r_i^{(d)} = P(Y_d = i \,|\, Y_d > 0) \quad \text{for } i = 0, 1, \ldots .$$

Obviously,

$$q_0^{(d)} = 1 - \left(\frac{5}{6}\right)^d, \quad \text{and} \quad q_i^{(d)} = \left(\frac{5}{6}\right)^d r_i^{(d)} \quad \text{for } i, d = 1, 2, \ldots ,$$

and

$$r_i^{(d)} = \sum_{j=2}^{6} \frac{1}{5} r_{i-j}^{(d-1)} \ , \ i = 2d, 2d+1, \cdots, 6d, \quad \text{and} \quad r_i^{(d)} = 0 \text{ otherwise} ,$$

with the convention $r_0^{(0)} = 1$ and $r_i^{(0)} = 0$ for $i \neq 0$.

3 Two one-person games

To get some insight, let us consider the following two one-person games. In the first one you try to minimize the expected number of throws needed to reach G points. In the second one you maximize the probability of reaching G in a given number of throws.

3.1 Expected number of throws

Define $V(i)$ as the minimal expected number of throws needed to reach G when starting with i points. Then we have the ordinary dynamic programming equation (cf. [1]):

$$V(i) = \min_d \left\{ 1 + q_0^{(d)} V(i) + \sum_{j=2d}^{6d} q_j^{(d)} V(i+j) \right\}$$

or, equivalently,

$$V(i) = \min_d \left\{ \frac{1}{1-q_0^{(d)}} \left[1 + V(i) + \sum_{j=2d}^{6d} q_j^{(d)} V(i+j) \right] \right\} ,$$

where $V(i) = 0,\ i \geq G$.

Table 2 below gives the minimal expected number of throws and $d^*(i)$, the optimal number of dice to use in state i. As we see, the number to use varies quite a lot. Even using 7 dice is optimal in some states. Apparently the optimal strategy attempts to reach $G = 40$ in a certain number of 'successful' throws. In states 0 up to 11 this number is 2, whereas for $i \geq 12$ this number appears to be 1.

i	$V(i)$	$d^*(i)$	i	$V(i)$	$d^*(i)$	i	$V(i)$	$d^*(i)$	i	$V(i)$	$d^*(i)$
39	1.2000	1	29	2.0836	3	19	3.1929	6	9	4.5383	4
38	1.2000	1	28	2.1427	4	18	3.3010	6	8	4.6426	5
37	1.4400	2	27	2.2138	4	17	3.4355	6	7	4.7438	5
36	1.4400	2	26	2.3218	4	16	3.5919	6	6	4.8484	5
35	1.4880	2	25	2.4674	4	15	3.7622	6	5	4.9568	5
34	1.5840	2	24	2.5876	5	14	3.9239	7	4	5.0690	5
33	1.7376	3	23	2.6644	5	13	4.0539	7	3	5.1850	5
32	1.7664	3	22	2.7729	5	12	4.2027	7	2	5.3049	5
31	1.8259	3	21	2.9129	5	11	4.3254	4	1	5.4287	5
30	1.9277	3	20	3.0787	5	10	4.4300	4	0	5.5565	5

Table 2 Results for minimizing the expected number of throws for $G = 40$

3.2 Limited number of throws

Define $p^{(l)}(i)$ to be the maximal probability of reaching G in l throws, when starting with i points. Then, using DP, we have

$$p^{(l+1)}(i) = \max_d \left\{ \sum_j q_j^{(d)} p^{(l)}(i+j) \right\} ,$$

where $p^{(l)}(i) = 1$ for $i \geq G$, $l = 0, 1, \cdots$, and $p^{(0)}(i) = 0$ for $i < G$.

The results of a maximization with $G = 40$ and a limit L on the number of throws are given in Table 3. As we see, the number of dice to use is more regular, i.e., varies in a more monotonic way than in the case of minimizing the expected number of throws. You also see, that starting in 0 with 6 throws left you throw 4 dice. If the score turns out to be 0, you continue with 5 dice in the next throw. If then the score is 17 you continue with 5 dice again, but if it is 22 you continue with 4 dice, etc.

3.3 The game

The rules of the game state that in each throw simultaneously the two players have to decide upon the number of dice to use, so without seeing what the opponent is doing but knowing and using the scores so far. So, after a number of throws player 1 has reached a points and player 2 has reached b points. Thus the state space is two dimensional. If now player 1 decides to use k dice and player 2 uses l then the state changes from (a, b) into $(a + i, b + j)$ with probability $q_i^{(k)} q_j^{(l)}$.

The game is a stochastic terminating zero-sum game. If we assume that the number of dice to be used in each throw is limited by some number, D say ($D = 10$ in the TV game), then the game can be solved by dynamic programming recursively.

The value of the game is equal to the probability that player 1 wins minus the probability that player 2 wins, given that both players play optimally. Define

$$V(a,b) = \begin{cases} 1 & \textit{if} \quad a > b \quad \textit{and} \quad a \geq G; \\ 0 & \textit{if} \quad a = b \geq G; \\ -1 & \textit{if} \quad a < b \quad \textit{and} \quad b \geq G. \end{cases} \tag{1}$$

We want to determine $V(a, b)$ for both a and b less than G and the optimal, possibly randomized, actions that guarantee this value.

3.4 Randomized actions

The first question might be: do the players have to randomize the number of dice to use in a throw? Some insight is already gained by just looking at the

	$L = 1$		$L = 2$		$L = 3$		$L = 4$		$L = 5$		$L = 6$	
i	$P(i)$	d^*	$P(i)$	d^*	$P(i)$	d^*	$P(i)$	d^*	$P(i)$	d^*	$P(i)$	d^*
39	0.833	1	0.972	1	0.995	1	0.999	1	1.000	1	1.000	1
38	0.833	1	0.972	1	0.995	1	0.999	1	1.000	1	1.000	1
37	0.694	2	0.921	1	0.982	1	0.996	1	0.999	1	1.000	1
36	0.694	2	0.907	2	0.975	1	0.994	1	0.999	1	1.000	1
35	0.667	2	0.894	2	0.967	2	0.990	1	0.997	1	0.999	1
34	0.611	2	0.867	2	0.957	2	0.987	2	0.996	1	0.999	1
33	0.574	3	0.838	2	0.945	2	0.982	2	0.994	2	0.998	1
32	0.560	3	0.812	3	0.930	2	0.976	2	0.992	2	0.998	2
31	0.532	3	0.795	3	0.916	2	0.969	2	0.990	2	0.997	2
30	0.486	3	0.765	3	0.898	3	0.960	2	0.986	2	0.995	2
29	0.471	4	0.743	3	0.885	3	0.952	2	0.982	2	0.994	2
28	0.455	4	0.716	3	0.868	3	0.941	3	0.977	2	0.992	2
27	0.428	4	0.693	4	0.851	3	0.932	3	0.972	2	0.989	2
26	0.395	5	0.667	4	0.830	3	0.921	3	0.965	2	0.986	2
25	0.386	5	0.649	4	0.812	3	0.909	3	0.959	3	0.983	2
24	0.370	5	0.625	4	0.793	4	0.897	3	0.952	3	0.979	2
23	0.346	5	0.597	5	0.773	4	0.883	3	0.944	3	0.975	2
22	0.325	6	0.577	5	0.755	4	0.869	3	0.936	3	0.970	3
21	0.316	6	0.561	5	0.738	4	0.855	3	0.927	3	0.965	3
20	0.302	6	0.539	5	0.716	4	0.839	4	0.917	3	0.959	3
19	0.281	6	0.510	5	0.691	4	0.822	4	0.905	3	0.953	3
18	0.268	7	0.494	6	0.675	5	0.808	4	0.895	3	0.946	3
17	0.259	7	0.480	6	0.659	5	0.794	4	0.884	3	0.939	3
16	0.246	7	0.460	6	0.639	5	0.778	4	0.872	3	0.931	3
15	0.229	7	0.436	6	0.616	5	0.759	4	0.858	4	0.923	3
14	0.220	8	0.421	7	0.598	5	0.742	4	0.846	4	0.914	3
13	0.213	8	0.408	7	0.581	5	0.726	4	0.833	4	0.905	3
12	0.202	8	0.391	7	0.563	6	0.708	5	0.820	4	0.895	3
11	0.187	8	0.371	7	0.542	6	0.690	5	0.805	4	0.884	3
10	0.182	9	0.356	8	0.525	6	0.674	5	0.792	4	0.874	4
9	0.175	9	0.345	8	0.510	6	0.659	5	0.778	4	0.864	4
8	0.165	9	0.331	8	0.493	6	0.642	5	0.763	4	0.853	4
7	0.154	10	0.314	8	0.473	6	0.623	5	0.747	4	0.841	4
6	0.150	10	0.301	8	0.457	7	0.607	5	0.732	4	0.829	4
5	0.144	10	0.291	9	0.443	7	0.591	5	0.717	5	0.817	4
4	0.136	10	0.278	9	0.428	7	0.574	5	0.702	5	0.804	4
3	0.126	10	0.264	9	0.411	7	0.557	6	0.687	5	0.791	4
2	0.115	10	0.248	9	0.392	7	0.538	6	0.670	5	0.778	4
1	0.102	10	0.230	10	0.373	7	0.520	6	0.653	5	0.763	4
0	0.088	10	0.214	10	0.355	7	0.501	6	0.636	5	0.749	4

Table 3 Maximal probability of reaching $G = 40$ in at most L throws.

game starting in $(G-1, G-1)$. Knowing that the value of this symmetric state has to be zero we can check wether there is a 'deterministic move' (i.e., using a fixed number of dice) that guarantees the value 0. If there would be an optimal deterministic throw, then we must have for some $d \le D$, where D is the maximal number of dice that can be thrown, and for all l

$$V^{(d,l)}(G-1, G-1) := \frac{1}{1 - q_0^{(d)} q_0^{(l)}} \sum_{i,j;\ i+j>0} q_i^{(d)} q_j^{(l)} V(G-1+i, G-1+j) \ge 0 \,.$$

Computing $\min_l V^{(d,l)}(G-1, G-1)$ for all d leads to the results in Table 4.

d	$\min_l V^{(d,l)}(G-1, G-1)$	best response to d
1	-0.3951	2
2	-0.2282	3
3	-0.1282	4
4	-0.0649	5
5	-0.1072	1
6	-0.2467	1
7	-0.3656	1
8	-0.4666	1
9	-0.5522	1
10	-0.6245	1

Table 4 Best result for player 1 restricting to deterministic moves.

So, there is no optimal number of dice. The best number is 4, but even then the best you can get is -0.0649. If your opponent knows the number of dice you use, it is optimal for him to use one dice more, unless you use 5 or more dice, then his optimal choice is 1. Thus, randomization is necessary.

4 The stochastic game

The two-person zero-sum stochastic game is in fact a terminating, even contracting game. In each move (throw of the two players) the state of the game gets closer to the payoff-zone: the set of states (a, b) with $\min\{a, b\} \ge G$. (Define the distance from (a, b) to the payoff-zone as $2G - a - b$ if both a and b less than G. Then with a probability of at least $1 - (q_0^{(D)})^2$ the distance decreases by at least 2.)

The value of the game and the optimal moves of the two players can be computed by repeatedly solving the appropriate matrix games.

Let $x = (x_1, x_2, \ldots, x_D)$ be a randomized move for player 1, i.e., player 1 throws d dice with probability x_d where $\sum_d x_d = 1$. The first approach to think off is to recursively compute $V(a, b)$ via a sequence of LP-problems, starting in $(a, b) = (G-1, G-1)$ and working backwards, step by step, until $(a, b) = (0, 0)$. This requires to solve the optimization problem:

$$\max V$$

$$\textit{subject to}$$

$$\sum_d x_d \left(\sum_{i+j>0} q_i^{(d)} q_j^{(l)} V(a+i, b+j) + q_0^{(d)} q_0^{(l)} V \right) \geq V, \quad l = 1, \ldots, D,$$

$$x_d \geq 0, \quad d = 1, \ldots, D, \quad \sum_d x_d = 1 \ ,$$

where, for $i + j > 0$, the values $V(a + i, b + j)$ have been computed before and hence are known. (V is unrestricted in sign.) However, this optimization problem is not exactly an LP-problem because of the nonlinear term $\sum_d x_d q_0^{(d)} q_0^{(l)} V$.

To make an LP-approach possible, we proceed as follows. Define $V^{(n)}(a, b)$ as the value of the game if it is played at most n times with a terminal reward 0, if the game has not reached the payoff-zone in n steps. Thus, $V^{(0)}(a, b) := 0$ if $a < G$ and $b < G$. Also, define

$$V^{(n)}(a, x, b, l) = \sum_d x_d \sum_{i,j} q_i^{(d)} q_j^{(l)} V^{(n-1)}(a+i, b+j), \ n > 0 \ ,$$

with the convention that, for $n \geq 0$ and $a \geq G$ or $b \geq G$, $V^{(n)}(a, b) = V(a, b)$ with $V(a, b)$ as defined in (1).

Then in iteration n in state (a, b) the value of the game and the (an) optimal move for player 1 can be obtained from the following LP-problem (cf. [3]):

Matrix game

$$\max V$$

$$\textit{subject to}$$

$$V^{(n)}(a, x, b, l) \geq V, \quad l = 1, \ldots, D,$$

$$x_d \geq 0, \quad d = 1, \ldots, D, \quad \sum_d x_d = 1.$$

The optimal value V satisfies $V = V^{(n)}(a, b)$ and the optimal $x^{(n)}(a, b)$ is the (an) optimal move for player 1 in state (a, b) in iteration n. $V^{(n)}(a, x, b, l)$ converges exponentially fast to the value of the game, and $x^{(n)}$ is nearly optimal for n sufficiently large.

Similarly, we can compute a (nearly) optimal strategy for player 2. Of course, for symmetry reasons the optimal move for player 2 in (a, b) is the same as the optimal move for player 1 in (b, a).

Remark 1 In order to profit from the contracting properties of the dynamic programming scheme for V^n, one may introduce a so-called weighted supremum norm μ. Defining $\mu(a,b) = \alpha^{a+b}$ for some $\alpha < 1$ the model will be contracting with respect to the μ-norm and nearly optimal strategies and upper and lower bounds can be computed from the difference between $V^{(n+1)}$ and $V^{(n)}$, cf. [4].

4.1 The optimal strategy

In Table 5 we present some results for the optimal strategy for the case the maximum number of dice D is equal to 5. The table should be read as follows. If, for instance, player 1 has $G-1$ points and player 2 has $G-3$ points, then player 1 will use 2, 4 or 5 dice with probabilities 0.172, 0.151 and 0.677 respectively.
Our results have also shown that for smaller values than $G-13$ the players use non-randomized decisions only.
What we also see, for instance in state $(G-5, G-13)$ player 1 will use 4 dice and player 2 will use 5 dice. So both players use more dice then needed to reach the payoff-zone in order to beat the other player in case none of them throws a 1.

5 Variants

There are various modifications of this game possible. To mention a few.

1. Player 2 uses the optimal strategy with respect to one of one-person games discussed before. What is the optimal response for player 1 and how does this increase 'his value'? This 'game' can still be solved by ordinary DP.
2. Suppose at the start a coin is flipped to decide which player may start. Then, alternatingly they throw a number of dice until one of the players reaches G. Again simple DP suffices to obtain the optimal strategy.
3. Suppose when a player throws a 1 not only his score is 0, but he also loses all (or some of) the points collected so far.
4. Suppose the players know the outcomes of their own throws, but don't know what the other player has been doing at all. This is a game with imperfect information. Is it possible to determine an optimal strategy?
5. Suppose that in addition to the previous situation you also know how many dice your opponent has used. This too is a game with imperfect information.

References

1. Derman, C., Finite State Markovian Decision Problems, Academic Press, New York, 1970.

	G-13	G-12	G-11	G-10	G-9	G-8	G-7	G-6	G-5	G-4	G-3	G-2	G-1
	0	0	0	0	0	0	0	0	0	0	0	0	0
	0	0	0	0	0	0	0	0	0	0	0	0	0
G-13	0	0	0	0	0	0	0	0	0	0	0	0	0
	0	0	0	0	0	0	0	0	0	0	0	0	0
	1	1	1	1	1	1	1	1	1	1	1	1	1
	0	0	0	0	0	0	0	0	0	0	0	0	0
	0	0	0	0	0	0	0	0	0	0	0	0	0
G-12	0	0	0	0	0	0	0	0	0	0	0	0	0
	0	0	0	0	0	0	0.144	0	0	0	0.238	0	0
	1	1	1	1	1	1	0.856	1	1	1	0.762	1	1
	0	0	0	0	0	0	0	0	0	0	0	0	0
	0	0	0	0	0	0	0	0	0	0	0	0	0
G-11	0	0	0	0	0	0	0	0	0	0	0	0	0
	0	0	0	0	0	0	0.366	0.140	0	0	0	0.233	0
	1	1	1	1	1	1	0.634	0.860	1	1	1	0.767	1
	0	0	0	0	0	0	0	0	0	0	0	0	0
	0	0	0	0	0	0	0	0	0	0	0	0	0
G-10	0	0	0	0	0	0	0	0.203	0.089	0	0	0.268	0.156
	0	0	0	0	0	0	0	0	0	0	0	0	0
	1	1	1	1	1	1	1	0.797	0.911	1	1	0.732	0.844
	0	0	0	0	0	0	0	0	0	0	0	0	0
	0	0	0	0	0	0	0	0	0	0	0	0	0
G-9	0	0	0	0	0	0	0	0.277	0.196	0.085	0	0	0.259
	0	0	0	0	0	0	0	0	0	0	0	0	0
	1	1	1	1	1	1	1	0.723	0.804	0.915	1	1	0.741
	0	0	0	0	0	0	0	0	0	0	0	0	0
	0	0	0	0	0	0	0	0	0	0	0	0	0
G-8	0	0	0	0	0	0	0	0.335	0.272	0.192	0.083	0	0.330
	0	0	0	0	0	0	0	0	0	0	0	0	0
	1	1	1	1	1	1	1	0.665	0.728	0.808	0.917	1	0.670
	0	0	0	0	0	0	0	0	0	0	0	0	0
	0	0	0	0	0	0	0	0	0	0	0	0	0
G-7	0	0	0	0	0	0	0	0	0.332	0.269	0.190	0.082	0
	1	0.091	0.006	0	0	0	0	0	0	0	0	0	0
	0	0.909	0.994	1	1	1	1	1	0.668	0.731	0.810	0.918	1
	0	0	0	0	0	0	0	0	0	0	0	0	0
	0	0	0	0	0	0	0	0	0.226	0.209	0.198	0.172	0.076
G-6	0	0	0	0	0	0	0	0	0	0.073	0.042	0	0
	1	1	0.178	0.101	0.093	0.008	0	0	0	0	0	0	0
	0	0	0.822	0.899	0.907	0.992	1	1	0.774	0.718	0.760	0.828	0.924
	0	0	0	0	0	0	0	0	0	0	0	0	0
	0	0	0	0	0	0	0	0	0	0.202	0.196	0.187	0.166
G-5	0	0	0	0	0	0	0	0.016	0	0.058	0.081	0.046	0
	1	1	1	0.260	0.260	0.183	0.054	0	0	0	0	0	0
	0	0	0	0.740	0.740	0.817	0.946	0.984	1	0.740	0.723	0.767	0.834
	0	0	0	0	0	0	0	0	0	0	0	0	0
	0	0	0	0	0	0	0	0.064	0.127	0.176	0.196	0.190	0.182
G-4	0	0	0	0	0	0	0	0	0.043	0.053	0.064	0.084	0.048
	1	1	1	1	0.445	0.358	0.229	0.100	0	0	0	0	0
	0	0	0	0	0.555	0.642	0.771	0.836	0.830	0.771	0.740	0.726	0.770
	0	0	0	0	0	0	0	0	0	0	0	0	0
	0	0	0	0	0	0	0	0.064	0.137	0.146	0.176	0.196	0.190
G-3	1	0.001	0	0	0	0	0	0	0	0.075	0.053	0.065	0.084
	0	0.999	1	1	1	0.556	0.408	0.227	0.134	0	0	0	0
	0	0	0	0	0	0.444	0.592	0.709	0.729	0.779	0.771	0.739	0.726
	0	0	0	0	0	0	0	0	0	0	0	0	0.073
	0	0	0	0	0	0	0	0	0.142	0.172	0.146	0.176	0.067
G-2	1	1	0.070	0.015	0	0	0	0	0	0	0.075	0.053	0.172
	0	0	0.930	0.985	1	1	0.612	0.577	0.204	0.151	0	0	0
	0	0	0	0	0	0	0.388	0.423	0.654	0.677	0.779	0.771	0.688
	0	0	0	0	0	0	0	0	0	0	0	0.030	0
	0	0	0	0	0	0	0	0	0	0.180	0.172	0.171	0.176
G-1	1	1	1	0.162	0.143	0.059	0	0	0	0	0	0	0.053
	0	0	0	0.838	0.857	0.941	1	0.817	0.760	0.194	0.151	0.168	0
	0	0	0	0	0	0	0	0.183	0.240	0.626	0.677	0.631	0.771

Table 5 Optimal strategy for player 1 in (k, l) with $G - 13 \leq k, l \leq G - 1$

2. Hordijk,A., Dynamic Programming and Markov Potential Theory, Mathematical Centre tracts 54, 1974.
3. Maitra, A. and D. Sudderth, Discrete Gambling and Stochastic Games, Springer-Verlag, Berlin, 1996.
4. Van der Wal, J. and J. Wessels, Successive approximation methods for Markov games, in: Markov Decision Theory, Mathematical Centre tracts 93 (eds. H.C. Tijms and J.Wessels, pp. 39-55, 1977.